1A

УДК 681.324

Р.К. Алекперов
Институт информационных технологий НАН Азербайджана, г. Баку

Организация распределенных вычислений на базе GRID-технологии

В данной работе рассмотрена организация распределенных вычислительных сред (РВС) на основе GRID-технологии. Показано, что при создании РВС на основе GRID-технологии надо учитывать такие факторы, как архитектура, производительность компьютеров, скорость передачи каналов связи, разнородность подзадач по вычислительной сложности, оптимальное распределение заданий, интеграция вычислительных ресурсов компьютеров разных организаций и т.д.

Введение

Как известно, повышение производительности вычислительных систем всегда было и остается актуальной проблемой. История развития информационно-вычислительных технологий показывает, что, с одной стороны, увеличения производительности компьютеров возможно достичь за счет повышения тактовой частоты процессора, а с другой – за счет применения методов, ориентированных на обеспечение параллелизма. Параллельные вычисления играют большую роль в решении сложных задач [1], [2].

В настоящее время все больше возрастает потребность в решении таких актуальных и трудоемких задач, как предсказание погоды, моделирование физических процессов аэродинамических свойств объектов и др. Многие из них относятся к задачам, решение которых возможно путем применения методов параллельной обработки данных [3], [4].

Цель работы – определить организацию распределенных вычислительных сред на основе GRID-технологии и проанализировать специфические особенности этих задач.

Для решения больших сложных задач традиционно используются специализированные многопроцессорные системы [5]. Данные системы обладают четко определенным составом и структурой, что, с одной стороны, обеспечивает простоту их использования, но с другой – они достаточно дороги.

Быстрое развитие информационно-вычислительных технологий и сетевых средств в последнее время привело к увеличению вычислительной мощности, сосредоточенной в компьютерных сетях.

Никакая вычислительная система не может сравниться ни по пиковой производительности, ни по объему оперативной или дисковой памяти с теми суммарными ресурсами, которыми обладают компьютеры, подключенные к глобальным сетям. Сейчас в мире насчитывается множество компьютеров, в том числе рабочие станции, ПК, мощный сервер и кластеры, суперкомпьютерные системы. Многие из них объединены в сети. Благодаря различным службам, таким, как e-mail, ftp, www, в Интернете возможно обмениваться информацией между компьютерами. С развитием сети Интернет появилась концепция использования вычислительных ресурсов географически распределенных вычислительных систем, в том числе обычных персональных компьютеров, для решения сложных задач. Такая концепция получила название метакомпьютинг или GRID [5], [6].

Анализ организации распределенных вычислений в сетевой среде

GRID – географически распределенная вычислительная система, объединяющая множество разных ресурсов (процессорные, коммуникации, система хранения данных, информационные системы, а также программные фонды), доступ к которым пользователь может получить из любой точки, независимо от места их расположения. GRID также сетевая служба, и подобно тому, как происходит обмен данными между подключенными к Интернету компьютерами, так и она позволяет обмениваться вычислительными ресурсами, дисковым пространством и т.д. Основная задача GRID – создание протоколов и сервисов для обеспечения надежного и безопасного доступа к географически распределенным информационным вычислительным ресурсам – отдельным компьютерам, кластерам, суперкомпьютерным центрам, хранящим информацию, сетям и т.д. [7-9].

Концепция GRID-систем возникла на базе успехов, достигнутых, прежде всего, в следующих направлениях [10]:

– резкого повышения производительности микропроцессоров массового производства;

– появления высокоскоростных оптоволоконных линий связи;

– феномена WWW/internet-глобализации процесса обмена информацией и интеграции мировой экономики;

– развития методов организации решения сложных вычислительных задач;

– совершенствования технологии и средств информационной безопасности.

В отличие от традиционной вычислительной среды, GRID-система имеет целый набор присущих ей особенностей:

1) GRID-система обладает огромными ресурсами, которые не сравнимы с ресурсами обычных компьютеров;

2) GRID-система является распределенной по своей природе. Компоненты GRID-системы могут быть удалены друг от друга на сотни и тысячи километров, что неизбежно вызовет большую латентность и, следовательно, скажется на оперативности их взаимодействия;

3) GRID-система может динамически менять конфигурацию. Какие-то компьютеры к ней подсоединяются и делегируют права на использование своих ресурсов, какие-то отключаются и становятся недоступными. Но для пользователя работа с GRID-системой должна быть прозрачной. Задача системы поддержки работы GRID-системы состоит в поиске подходящих ресурсов, проверке ее работоспособности, распределении поступающих задач вне зависимости от текущей конфигурации GRID-системы в целом;

4) GRID-система неоднородна. При распределении заданий нужно учитывать особенности операционных систем, входящих в ее состав. Разные системы поддерживают различные системы команд и форматы представления данных. Различные системы в разное время могут иметь различную загрузку, связь с вычислительными системами идет по каналам с различной пропускной способностью. Наконец в состав GRID-системы могут входить системы с принципиально различной архитектурой, начиная от персональных компьютеров и заканчивая мощнейшими вычислительными системами;

5) GRID-система объединяет ресурсы различных организаций. Политика доступа и использования конкретных ресурсов может сильно меняться в зависимости от их принадлежности к той или иной организации. Поэтому политика его администриро-

вания может быть определена лишь в самых общих чертах. Вместе с тем согласованность работы огромного числа составных частей GRID-системы предполагает обязательную стандартизацию работы всех ее служб и сервисов. Несмотря на кажущуюся нереальность создаваемой глобальной вычислительной системы, область применения GRID-системы обширна.

Изначально GRID-технологии предназначались для решения сложных научных, производственных и инженерных задач, которые невозможно решить в разумные сроки на отдельных вычислительных машинах. Однако теперь область применения технологий GRID не ограничивается только этими типами задачами. По мере своего развития GRID проникает в промышленность и бизнес, крупные предприятия создают свои GRID для решения собственных производственных задач. Таким образом, GRID претендует на роль универсальной инфраструктуры для обработки данных, в которой функционирует множество служб (Grid Services), которые позволяют решать не только конкретные прикладные задачи, но и предлагают сервисные услуги: поиск необходимых ресурсов, сбор информации о состоянии ресурсов, хранение и доставку данных [11].

Область применения GRID сейчас охватывает ядерную физику, защиту окружающей среды, предсказание погоды и моделирование климатических изменений, численное моделирование в машиностроении и авиастроении, биологическое моделирование, фармацевтику [12].

Кроме того, применение GRID может дать новое качество решения следующих классов задач:

– массовая обработка потоков данных большого объема;

– многопараметрический анализ данных;

– моделирование на удаленных суперкомпьютерах;

– математическое моделирование с высокоточной визуализацией вычислительных экспериментов;

– сложные бизнес-приложения с большими объемами вычислений.

Таким образом, GRID служит универсальной и эффективной инфраструктурой для высокопроизводительных распределенных вычислений и обработки данных. Однако самая важная роль концепции GRID состоит не в том, чтобы увеличить производительность саму по себе, а в том, чтобы предложить решения новых задач, связанных с созданием надежных, масштабируемых и защищенных распределенных систем [13-16].

При решении задачи интеграции вычислительных ресурсов компьютеров разных организаций наиболее серьезными являются две проблемы [5]:

– сложность администрирования компьютеров, принадлежащих разным учреждениям, организациям, фирмам;

– рациональное использование так называемой низкокачественной вычислительной мощности, сосредоточенной в сети.

Сложность использования компьютеров, принадлежащих разным организациям, вполне объяснима. Даже при наличии желания администрации какой-либо организации предоставить простаивающую вычислительную мощность своих компьютеров для выполнения расчетов, обеспечить административный доступ для сторонних пользователей к операционным системам и файлам трудно.

С другой стороны, операционная система рабочей станции, используемой в качестве вычислительного узла GRID, конфигурируется совершенно конкретным образом в расчете на максимальную производительность. При работе на этом узле соблюдается строжайшая дисциплина доступа к его ресурсам из единого центра – центрального узла GRID. Таким способом достигаются управляемость и целостность системы. С другой

стороны, свободные ресурсы компьютеров, принадлежавшие разным организациям, используются не на принципах добровольности, а на традиционных рыночных принципах. Коммерциализация вычислений дает возможность увеличить количество участников в созидаемой GRID [2], [9].

Технологии, программные средства и требования для организации распределенных вычислений

В мире существует многочисленный класс задач, которые удается разбить на множество подзадач, не связанных между собой. Для выполнения такого множества подзадач задержки связи между компьютерами практически не имеют значения. От них зависит только время сбора подмножества результатов.

Следовательно, все большее количество сложных задач можно решать за приемлемое время с использованием КС. Наилучшим образом для решения на GRID подходят задачи комбинаторного и поискового типа, где решаемые подзадачи слабосвязанных или практически независимых друг от друга и основная часть работы на компьютерах выполняются в автономном режиме. Основная схема работы в этом случае примерно такая: специальный агент, расположенный на вычислительном узле (компьютере пользователя), определяет свободный вычислительный ресурс этого компьютера, соединяется с центральным узлом GRID и получает от него решаемые подзадачи. По окончании решения подзадачи вычислительный узел передает результаты центральным узлам GRID. Работая в такой среде, пользователь лишь выдает задание на решение задачи, а остальное центральный узел GRID делает сам: ищет доступные вычислительные ресурсы, отслеживает их работоспособность, осуществляет передачу данных, если требуется, выполняет преобразование данных в формат компьютера, на котором будет выполняться задача, и т.п. Пользователь может даже и не узнать, ресурсы какого именно компьютера были ему предоставлены. Если потребовались вычислительные мощности для решения задачи, то вы подключаетесь к GRID, выдаете задание и получаете результат.

Здесь существует почти полная аналогия с электрической сетью. Включая электрический чайник в розетку, вы не задумываетесь, какая станция производит электроэнергию. Вам нужен ресурс, вы им пользуетесь. По аналогии именно с электрической сетью распределенная вычислительная среда в англоязычной литературе получила название GRID.

Эффективность использования GRID-системы в решении сложных задач в значительной мере зависит от качества прикладных параллельных программ и архитектуры системы. Программа считается эффективной тогда, когда во время ее выполнения загружены все процессоры, выделенные под процессы. Но практически это нереализуемо. Существует множество факторов, вытекающих из технических возможностей организации распараллеливания процессов [18].

Гетерогенность состава вычислительных узлов и непредсказуемые изменения вычислительной среды во время решения задачи приводят к проблеме рационального использования вычислительной мощности, сосредоточенной в сети [9].

Анализ существующих на сегодняшний день технологий и программных средств, позволяющих решать вычислительные задачи в КС, показывает, что процесс разработки приложений для решения задач с использованием сети в качестве вычислительного ресурса является сложным, так как содержит множество этапов, начиная от разработки параллельного алгоритма и заканчивая мониторингом ресурсов и распределения нагрузки [10].

Проблема распределения нагрузки в параллельных вычислениях является одной из самых важных, особенно в такой динамично меняющейся среде, как КС. И именно от решения этой проблемы в основном зависит эффективность параллельного решения задачи, то есть тот выигрыш во времени, который можно получить по сравнению с последовательным решением.

Распараллеливание последовательной программы предполагает разбиение этой программы на процессы $P_i, i \in \{1,2,...,N\}$, где N – число процессов. Для каждого процесса имеются множество его входных данных I_i и множество выходных (вырабатываемых процессом P_i) данных O_i. Два процесса P_1 и P_2 могут выполняться независимо и параллельно, если между ними нет зависимости по данным: $I_1 \cap O_2 = \varnothing$ и $I_2 \cap O_1 = \varnothing$. Эти условия гарантируют, что выходные данные процесса P_2 не могут изменять содержание входных данных процесса P_1 и, соответственно, выходные данные процесса P_1 не могут изменять содержание входных данных процесса P_2. Эти требования могут быть распространены на произвольное число процессов, которые могут выполняться независимо и параллельно. Распределение задачи на независимые подзадачи зависит от того, сколько процентов решаемых задач можно распараллеливать. Обозначим через f ($0 \le f \le 1$) долю операции, которая выполняется последовательно (в абсолютных единицах), при этом если $f=1$, то программа выполняется последовательно, а если $f=0$, то программа полностью может распараллеливать. Для того чтобы оценить, какое ускорение S может быть получено на компьютере из P процессоров при данном значении f, можно воспользоваться законом Амдала [2].

$$S \le 1/(f + \frac{1-f}{P}).$$

Из формулы следует, что P-кратное ускорение может быть достигнуто только для полностью независимых программ.

Одной из основных задач в области параллельного программирования является накопление опыта переносимости параллельных программ. Вплоть до недавнего времени программы, созданные для одной системы, не могли быть выполнены на другой, что препятствовало накоплению знаний о параллельных алгоритмах.

Преодоление этого стало возможно путем введения стандартизаций программы в параллельных вычислениях, таких, как PVM (параллельная виртуальная машина) [19], MPI (интерфейс передачи сообщений) [20], Open MP (интерфейс прикладных программ для разделяемой памяти) и, возможно, других. Эти стандартные программы должны обеспечить переносимость программ между любыми параллельными системами, начиная от, например, сети Ethernet рабочих станций до MPP-системы и глобальной сети суперкомпьютеров различной архитектуры.

Для создания GRID используется различное программное обеспечение (GLOBUS TOOLKIT, CORBA, DCOM и др.). Наиболее широко используется программное обеспечение GLOBUS TOOLKIT. GLOBUS TOOLKIT – набор программного обеспечения и служб, которые являются базисом для построения GRID и приложений для нее. Последняя версия этого пакета GLOBUS TOOLKIT 3 (GT3) включает средства разработки служб в соответствии со стандартом OGSI (OPEN GRID SERVICES INFRASTRUCTURE), среду их функционирования. К основным видам сервисов, включенных в GLOBUS TOOLKIT, можно отнести следующие: связь, управление ресурсами, безопасность, информационное обслуживание, доступ к удаленным данным, запуск и управление зада-

ниями. GLOBUS TOOLKIT признан ведущими производителями программного обеспечения стандартом де-факто для GRID [21].

Для создания GRID-системы с помощью сетевых приложений обычно используется распределенная архитектура (клиент – сервер), при которой сервер-центральный компьютер выдает задания различным клиентам-компьютер [22], [23]. Для решения больших комбинаторных задач, как правило, используется GRID-система одноуровневой архитектуры. Система состоит из двух основных компонентов: центральный компьютер и узлы. Центральный компьютер – это центральная часть системы, отвечающая за разделение исходной задачи на подзадачи, распределение подзадачи, координацию работ всех узлов, контроль целостности результата, сбор результата расчета в единое целое. Узел – любая вычислительная единица (ПК, рабочая станция, узел кластера, виртуальная машина), в которой происходит основной расчет прикладной программы. Отдельные разделенные подзадачи передаются с центрального компьютера на узлы, где происходит расчет, результаты которого передаются обратно на центральный компьютер. Узлы отвечают за расчет блоков прикладной задачи, запрос заданий для расчета от центрального компьютера, передачу результатов расчета на центральный компьютер. Например, в задаче по расшифровке фразы, закодированной с 64-битным ключом, используется одноуровневая архитектура GRID-системы. Основное требование здесь, чтобы задача позволяла разбиение на небольшие блоки достаточного размера с фиксированным количеством операций, чтобы компьютеры делали вычисления несколько часов. Настройка одноуровневой системы предельно простая и быстрая. Определяем количество компьютеров для расшифровки фразы с 64-битным ключом. Для нахождения нужного ключа надо проверить 2^{64} ключей. Количество ключей приблизительно равно $2^{64} \approx 16\times10^{18}$ комбинаций.

Допустим, что компьютерная сеть однородна, узлами ее являются персональные компьютеры Pentium IV с производительностью один миллиард операций в секунду. Персональные компьютеры соединяются с центральным компьютером высокоскоростной линией связи. Персональный компьютер Pentium IV с производительностью один миллиард операций в секунду непрерывно может проверять в сутки $8,64\times10^{13}$ (24 часов$\times10^{9}$) комбинаций. Для определения нужного ключа из 16×10^{18} ключей за одни сутки нам нужны $16\times10^{18} / 8,64\times10^{13} = 1,85\times10^{5}$ =185000 Pentium IV, что в пределах одной организации практически нереализуемо. Поэтому, чтобы получить заданные вычислительные мощности, необходимо использовать GRID-системы. Для организации более эффективных и надежных GRID-систем может быть использована многоуровневая архитектурная структура.

Большое количество компьютеров и каналов связи, используемых в одноуровневой GRID-системе, уменьшает безопасность и надежность системы. Для того чтобы устранить указанные недостатки, используют многоуровневую GRID-систему. В этих системах решаемые задачи одновременно разделяются на подзадачи и для каждого уровня определяется набор подзадач.

Результаты, полученные в нижних уровнях, используются в последующих уровнях. Компьютеры, используемые на нижних уровнях, используются и на верхних уровнях. Для поиска простых чисел Мерсина, решения системы алгебраических линейных уравнений и т.д. используются многоуровневые GRID-системы.

В многоуровневой GRID-системе показатели надежности относительно высоки за счет уменьшения количества используемых компьютеров и каналов связи [23].

В настоящее время в мире проводятся различные проекты по использованию вычислительных мощностей в КС, к числу которых можно отнести: SETI@home, DISTRI-

BUTED.NET, GIMPS, ANTHRAX, GLOBUS, CONDOR, CERN и др. Рассмотрим некоторые реальные проекты и существующие технологии GRID-системы.

Проект SETI@home (SEARCH FOR EXTRATERRESTRIAL INTELLIGENCE) – поиск внеземных цивилизаций с помощью распределенной обработки данных, поступающих с радиотелескопа. Присоединиться к проекту может любой желающий, загрузив на свой компьютер программу обработки радиосигналов. Доступны клиентские программы для WINDOWS, MACINTOSH, UNIX, OS/2. С момента старта проекта в мае 1999 года до мая 2002 года для участия в проекте зарегистрировались более 3,7 млн человек. Добровольцы, участвующие в проекте, получают порциями по 384 Кб радиоастрономических данных в файлах. Эти данные обрабатываются в фоновом режиме программой, загруженной с web-сайта SET@home.com. Она выполняет довольно объемные вычисления, связанные с быстрым преобразованием Фурье, но какие именно – для добровольца остается неизвестным. На компьютере G3 POWER MACINTOSH обработка одной порции (подзадачи) занимает порядка двух суток, в процессе работы компьютер пользователя имеет обратную связь, свидетельствующую о ходе выполнения процесса. По окончании результаты расчетов возвращаются в SET@home.com. Согласно приводимой на сайте статистике, суммарная производительность задействованных в проекте компьютеров во много раз превосходит производительность всех компьютеров из списка Top 500.

DISTRIBUTED.NET также является одним из самых больших объединений пользователей Интернета, предоставляющих свои компьютеры для решения сложных задач в распределенном режиме. Эти проекты связаны с задачами определения шифров (RSA CHALLENGES). С момента начала проекта в нем зарегистрировались около 200 тыс. человек.

Проект GIMPS (GREAT INTERNET MERSENNE PRIME SEARCH) ставит задачу поиска простых чисел Мерсена, то есть числа вида $2^P – 1$, где p является простым числом. В ноябре 2001 года в рамках данного проекта было найдено максимальное на данное время число Мерсена $2^{13466917} – 1$. Десятки тысяч компьютеров по всему миру, отдавая часть своих вычислительных ресурсов, работали над этой задачей два с половиной года.

В проекте ANTHRAX (Сибирская язва), который возник в связи с имевшимися в США случаями распространения по почте спор сибирской язвы и был направлен на поиск противоядия. Проект был запущен после того, как была выделена ключевая белковая компонента, спор. Вычислительная часть задачи заключалась в том, чтобы произвести на выделенном белке скрининг 3,57 млрд потенциальных ингибиторов токсина. Задача решалась на специальном варианте платформы GRID MP компании UNITED DEVICES. В распределенные вычисления было вовлечено около 1,9 млн серверов и ПК. Высокая точность и качество обеспечивались пятикратным уровнем избыточности при скрининге каждой молекулы. Полный скрининг был закончен за 24 дня. По свидетельству специалистов Оксфордского университета, если бы эта работа делалась традиционными методами, она бы длилась несколько десятков лет, а не четыре недели.

Несмотря на использование идеологии GRID-технологии, естественно, что имеется ряд проблем, задерживающих ее применение в широком масштабе.

Рациональное использование ресурсов GRID-системы определяется нижеперечисленными критериями [24]:

– необходимость использования всех доступных вычислительных ресурсов;

– минимизация простоев вычислительных узлов и обеспечение их постоянной загрузкой;

– минимизация накладных расходов;

– обеспечение безопасности;

– обеспечение надежности.

На основании данных критериев можно сделать оценку эффективности методов решения задач в сети и максимально выгодно использовать доступные ресурсы.

Заключение

Проведенный анализ показывает, что при создании РВС на основе КС приходится учитывать множество факторов: архитектуру, надежность, безопасность, производительность компьютеров, скорость передачи каналов связи, интеграцию вычислительных ресурсов компьютеров разных организаций, оптимальное распределение заданий и т.д. При этом главная цель заключается в повышении эффективности использования простаивающих вычислительных ресурсов КС.

Литература

1. Параллельные вычисления и задачи управления. (Аналитический обзор) / А.Л. Бунич, К.С. Гинсбург, А.В. Добровидов [и др.] // Автоматика и телемеханика. – 2002. – № 12. – С. 3-23.
2. Воеводин В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. – Санкт-Петербург : БХВ – Петербург, 2002. – 608 с.
3. Гери М. Вычислительные машины и труднорешаемые задачи / М. Гери, Д. Джонсон. – М. : Мир, 1992. – 416 с.
4. Задачи для суперкомпьютеров. Режим доступа : http://parallel.ru/research/apps.html.
5. Корнев В.В. Вычислительные системы / Корнев В.В. – М. : Гелиос АРВ, 2004. – 512 с.
6. Черняк Л. Crid как будущее компьютинга / Л. Черняк // Открытые системы. – 2003. – № 1. – С. 16-19.
7. Демичев А.П. Введение в грид-технологии / А.П. Демичев, В.А. Ильин, А.П. Крюков // Препринт НИИЯФ МГУ. – 2007. – 85 с.
8. Управление заданиями в распределенной вычислительной среде / Коваленко В.И., Коваленко Е.И., Карягин Д.А. [и др.] // Открытые системы. – 2001. – № 5-6. – С. 22-28.
9. Foster I. The Grid: Blueprint for New Computing Infrastructure / I. Foster, C. Kesselman. – San Fransisko : Morgan Kaufman, 1999. – 438 p.
10. Крюков Ю.А. Вычислительная инфраструктура для прикладных задач, будущее и настоящее / Ю.А. Крюков // Геоинформатика. – 2004. – № 9. – С. 57-61.
11. Devid M. Model-Based Evaluation: From Dependability to Security / M. Devid, H. William, S. Kishor // IEEE Transactions on Dependable and Security Computinq. – January – March, 2004. – Vol. 1, № 1. – P. 48-65.
12. Methods and Experiences of Parallelizing Flood Models / L. Hluchy, V.D. Tran, D. Froehlich [etc.] // The 10th Euro PVM/MPI conference. LNCS 2840. – Sept. 2003. – P. 677-681.
13. Доменико Т. OGSA : где GRID встречается с Web / Т. Доменико // Открытые системы. – 2003. – № 1. – С. 47-50.
14. Малышкин Н.В. Метасистема для работы с удаленными вычислительными системами / Н.В. Малышкин // Распределенные и кластерные вычисления : 25-я школа-семинар. – Красноярск : Изд-во Института вычисл. моделир. СОРАН, 2002. – С.133-139.
15. Foster I. A Metacomputing Infrastructure Toolkit / I. Foster, C. Kesselman // International Journal of Applications. – 1997. – № 11. – P. 115-128.
16. Foster T. The Anatomy of the Grid : Enabling Scalable Virtual Organizations / T. Foster, C. Kasselman, S. Tuecke // International Journal of High Performance Computing Applications. – 2001. – № 15(3). – P. 200-222.
17. Структура и проблемы развития программного обеспечения среды распределенных вычислительных Грид / [Коваленко В.И., Коваленко Е.И., Карягин Д.А. и др.] // Препринт Института прикладной математики РАН. – 2002. – № 22. – С. 1-23.

18. Вересов И.Г. Адаптивные алгоритмы обработки информации в мультиагентных системах : автореф. дисс. на соискание ученой степени канд. тех. наук / И.Г. Вересов. – Санкт-Петербург, 2002. – 18 с.
19. PVM : Parallel Virtual Muchine, A Vser's Guide and Tutorial for Networked Parallel Computing / [Geist M., Begnelin A., Dongarra J. etc.]. – MIT Press, 1994. – P. 87-93.
20. Корнев В.Д. Параллельное программирование в MPI / Корнев В.Д. – Новосибирск : ИВМ и МГ СО РАН, 2002. – 84 с.
21. Кирьянов А.К. Введение в технологию Грид : [уч. пособие] / А.К. Кирьянов, Ю.Ф. Рябов. – Гатчина : ПИЯФ РАН, 2006. – 39 с.
22. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. Ван Стеен. – СПб. : Питер, 2003. – 877 с.
23. Филамофитский М.П. Система поддержки метакомпьютерных расчетов X-COM. Архитектура и технология работы / М.П. Филамофитский // Вычислительные методы и программирование. – 2004. – Т. 5, Раздел 2. – С. 1-9.
24. Алгулиев Р.М. Основные проблемы синтеза распределенных вычислительных сред на базе компьютерных сетей / Р.М. Алгулиев, Р.К. Алекперов // Телекоммуникации. – 2009. – № 10. – С. 5-9.

Р.К. Алекперов
Організація розподілених обчислень на базі GRID-технології
У даній роботі розглянута організація розподілених обчислювальних середовищ (РОС) на основі GRID-технології. Показано, що при створенні РОС на основі GRID-технології треба враховувати такі чинники, як архітектура, продуктивність комп'ютерів, швидкість передачі каналів зв'язку, різнорідність підзадач з обчислювальної складності, оптимальний розподіл завдань, інтеграція обчислювальних ресурсів комп'ютерів різних організацій і т.д.

R.K. Alekperov
GRID-Technology Based Organization of Distributed Computing
In this article the organization of DCE (Distributed Computing Environments) on the basis of GRID-technologies is considered. The analysis shows that creating DSC (Distributed Computing Systems) based on the GRID-technologies must take into account such factors as architecture, performance of computers, speed of communication channels, the diversity of subtasks on computational complexity, the optimal distribution of tasks, integration of computing resources of various organizations, etc.

Статья поступила в редакцию 02.11.2010.